



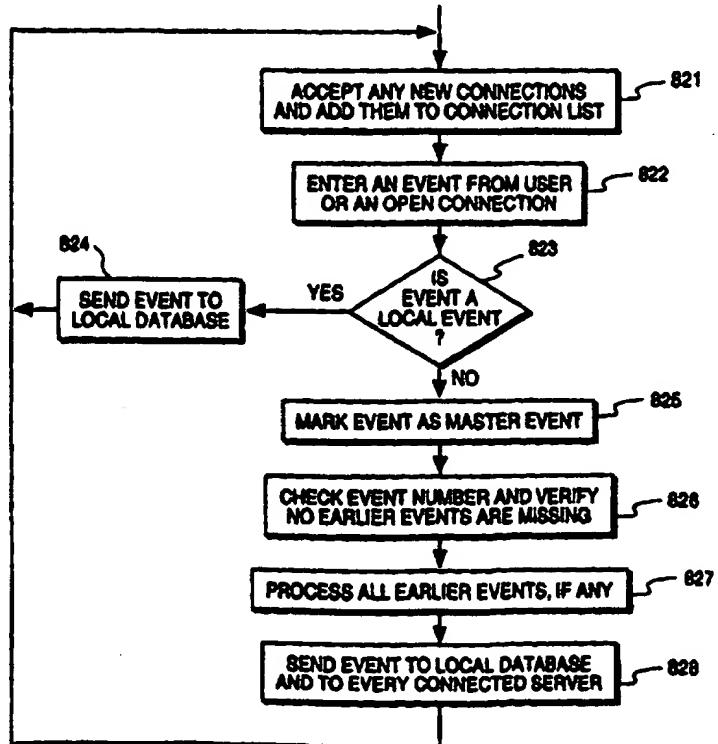
## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification <sup>6</sup> :  H04B		A2	(11) International Publication Number: <b>WO 97/47091</b>
(21) International Application Number: <b>PCT/US97/09054</b>		(43) International Publication Date: 11 December 1997 (11.12.97)	(43) International Publication Date: 11 December 1997 (11.12.97)
(22) International Filing Date: 22 May 1997 (22.05.97)		(81) Designated States: AU, CA, JP, KR, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).	
(30) Priority Data: 08/652,803 23 May 1996 (23.05.96) US		Published	Without international search report and to be republished upon receipt of that report.
(71) Applicant: ADOBE SYSTEMS INCORPORATED [US/US]; 345 Park Avenue, San Jose, CA 95110 (US).			
(72) Inventors: ROY, H. Scott; 1365 Green Street #304, San Francisco, CA 94109 (US). HARVEY, William, D.; Apartment 101, 432 High Street, Palo Alto, CA 94301 (US).			
(74) Agent: TROESCH, Hans, R.; Fish & Richardson P.C., Suite 100, 2200 Sand Hill Road, Menlo Park, CA 94025 (US).			

## (54) Title: REDUCING LATENCY WHEN SYNCHRONIZING ACCESS TO A MULTI-USER DATABASE OVER A NETWORK

## (57) Abstract

A method and apparatus for reducing network latency during execution of a multiple-player game across a computer network are provided. A master database represents a world model for the game, and a master event server sequences user inputs, or events, for updating the master database. The master event server and the master database are initially located on a particular node in the network. Every other node that is used by a player in the game has a slave event server and a slave database. Slave event servers are responsible for updating their local slave database, sending events from their local node to the master event server, and forwarding events to and from other slave servers. In the method, a determination is made of which player in the game has a role in the game requiring the least latency of any role and which node in the network is being used by that player. The master event server and the master database are then migrated to that node by changing the status of the slave server of that node to that of master event server. Once the master event server has been migrated, other slave servers can establish a direct connection with the new master event server.



**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FJ	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		

- 1 -

REDUCING LATENCY WHEN SYNCHRONIZING ACCESS  
TO A MULTI-USER DATABASE OVER A NETWORK

FIELD OF THE INVENTION

5 The present invention pertains to the field of computer networking. More particularly, the present invention relates to reducing latency when synchronizing access to a multi-user database over a network.

BACKGROUND OF THE INVENTION

10 Use of the Internet has been increasing rapidly in the past several years. One potential use of the Internet which has particular appeal for many users is the possibility of using the Internet to play computer games. Existing technology allows two players at 15 different locations to simultaneously play a game using a direct telephone or ISDN connection. Similarly, computer games can be implemented on a local area network (LAN) in which each player in the game uses a different computer. However, there are unique problems associated with 20 implementing a multi-player computer game across a wide area network (WAN), particularly one as expansive as the Internet.

One such problem is that even computer games which are designed for multiple players using different 25 computers tend to be sensitive to latency in communicating data between computers. Consider, for example, a computer game in which two or more players at remote locations interact with each other within a fictitious "game world." The game world can be modeled 30 by a database, which might be located on one of the player's computers or on a central server. The players interact with each other by entering inputs through a joystick, mouse, keyboard, or other input/output device. In order for the game to be enjoyable, the game world

- 2 -

must accurately reflect the current history of user inputs, or "events," received from all players. To resolve ambiguity of event ordering, events are typically collected at a central server that establishes the event sequence. However, differences in delays between the various computers involved in the game can hinder the proper sequencing of events when updating the game world. In addition, the speed with which events are communicated to and from the game world (i.e., the database) can be critical, depending upon the type of game being played. For example, in naturally fast-paced games, such as those involving close combat between characters and many simulations, high latency can cause the game to be unrealistic, if not completely unplayable. High latency can be quite problematic even in games in which only particular players have latency-sensitive roles.

Sensitivity to latency presents a significant obstacle to implementing certain multi-player computer games on the Internet, because the latencies associated with the Internet can be substantial and quite unpredictable. Various factors contribute to this unpredictability, including diverse delays between different communication paths (which may be caused by differences in communications hardware), uncertainty as to which physical route a given transmission will take, and wide variations in network communications traffic over time. Therefore, what is needed is a technique for overcoming such difficulties associated with the prior art when implementing multi-player computer games on a WAN. In particular, what is needed is a technique for resolving latency concerns in a multi-player game implemented on the Internet.

#### SUMMARY OF THE INVENTION

A method is provided of reducing network latency in accessing a database in a network. The network has a

- 3 -

number of nodes, including a first node and a second node. The first node includes a first event server for sequencing commands to the database, which is stored in one of the nodes in the network. The second node 5 includes a second event server for sequencing commands to the database. The method comprises the steps of changing the status of the first event server from master event server to slave event server based on a network latency criterion, and assigning the status of master event 10 server to the second event server based on the network latency criterion.

Other features of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

15 BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

20 Figure 1 illustrates a plurality of computer systems connected in a network configuration.

Figure 2 is a block diagram of a computer system in which the present invention can be implemented.

Figure 3 illustrates a logical configuration in 25 which a plurality of computer systems can be connected according to the present invention.

Figure 4A illustrates a single-player game based on a database running on a computer system.

Figure 4B illustrates a multi-player game running 30 in a network using a client-server configuration.

Figure 5A illustrates a multi-player game running in a network in accordance with the present invention, in which the master event server has not yet been migrated.

- 4 -

Figure 5B illustrates a multi-player game running in a network in which the master event server has been migrated.

Figure 6 illustrates a multi-player game running 5 in a network in which a slave event server has established a direct connection to the master event server.

Figure 7 is a flow diagram illustrating a technique for establishing a connection to a computer 10 system having multi-player game software.

Figure 8A is a flow diagram illustrating a technique used by a slave event server for routing events in a network.

Figure 8B is a flow diagram illustrating a 15 technique used by the master event server for routing events in a network.

Figure 9A is a flow diagram illustrating a technique used by the master event server for handling a slave migration event.

Figure 9B is a flow diagram illustrating a 20 technique used by a slave event server for handling a master migration event.

Figure 9C is a flow diagram illustrating a technique for determining whether the master event server 25 should be migrated to another computer.

Figure 10 is a flow diagram illustrating a technique used by a slave event server for changing a connection from a previous master event server to a current master event server.

30 DETAILED DESCRIPTION

A technique for reducing latency when synchronizing access to a multi-user database over a network is described. In the following description, for purposes of explanation, numerous specific details are

- 5 -

set forth in order to provide a thorough understanding of the present invention. It will be evident, however, to one skilled in the art that the present invention may be practiced without these specific details. In other 5 instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring the present invention.

The present invention includes methods having various steps. These steps may be embodied in computer 10 program instructions. The instructions can be used to cause a general purpose processor which is programmed with the instructions to perform the steps. Alternatively, the steps of the present invention may be performed by specific hardware components that contain 15 hardwired logic for performing the required steps, or by any combination of programmed general purpose computer components and custom hardware components.

The present invention is particularly suited to reducing latency when implementing a multi-player game on 20 the Internet. In particular, the present invention is suited to implementing a game in which a fictitious game world is modeled by a database, and in which one particular player of the game has a role with a latency requirement that is significantly more restrictive than 25 the other players' roles. An example of such a game is a simulated (American) football game in which one player plays offense and a second player plays defense at any given point in time. In such a game, the player on offense might be responsible for movement of the players, 30 including the speed and direction of various players, while the player on defense might be responsible only for selecting a predefined defensive pattern. Hence, in such a game, the player on offense would have substantial interaction with the game world, whereas the player on 35 defense would have little or no interaction with the game

- 6 -

world. Accordingly, the role of the player on offense would have a substantially more restrictive latency requirement than the role of the defensive player. Another example of a game in which one player has 5 substantially more restrictive latency requirement than the other players is a role-playing fantasy game. In such a game, one player acting as a "superhero" may participate on a very active level, while various parameters of the game might be determined at a high 10 level by one or a small number of players who participate only in a peripheral manner (e.g., by setting game conditions, selecting obstacles for the superhero to overcome, etc.). Hence, the player having the superhero role would have a much more restrictive latency 15 requirement than the other players, because he interacts with the database on a much more active level.

Referring now to Figure 1, a network configuration in which the present invention may be implemented is illustrated. A central server computer 1 is coupled via 20 the Internet 3 to a number of client computer systems 2. The server computer system 1 and each of the client computer systems 2 includes stored game software, including software in which the present invention is embodied. The game software includes various application 25 program interfaces (APIs) providing an interface to operating system software.

Figure 2 illustrates an example of an architecture which may be used for any of the client computer systems 2 or the server computer system 1. The architecture 30 includes a central processing unit (CPU) 10, memory 11, a mass storage device 12, a modem 13, and various input/output (I/O) devices 14. Memory 11 represents both random access memory (RAM) and read-only memory (ROM). Mass storage device 12 includes an optical, magnetic, or 35 other similar storage medium suitable for use with a

- 7 -

personal computer system. I/O devices 14 may include any combination of devices such as a mouse, a joystick, a trackball, a keyboard, a monitor, or other I/O devices. The software in which the present invention is embodied 5 may be stored in mass storage device 12, memory 11, or both. In addition, the software may be distributed between two or more of the computer systems illustrated in Figure 1.

Figure 4A shows a representation of a computer 10 system 40 running game software according to one prior art embodiment. A database 43 provides a world model for the game (i.e., a "game world"). User inputs 41 are received through a game-compatible I/O device 46 (such as a mouse or joystick) and provided to software 42. 15 Software 42 receives and sequences user inputs, or "events," that are used to update the database 43. Rendering commands 45 are sent from the database to a display 44 of the computer system 40 to generate various scenes representing the game world. Techniques for 20 implementing a game as illustrated in Figure 4A are well-known.

Figure 4B shows a well-known configuration for implementing a multi-user game on a network. In the configuration of Figure 4B, the game world is provided by 25 a database 53 located on a central server 51. Client computer systems 50 and 52 are used by Players A and B of the game, respectively. Software 54 located in the central server 51 receives events 58 from computers 50 and 52 and provides these events to the database 53 in 30 order to update the game world. Rendering commands 56 are then sent from the updated database 53 in the central server 51 across the network to update the displays 55 and 57 of client computers 50 and 52, respectively.

As mentioned above, implementing as shown in 35 Figure 4B on the Internet can be problematic if either of

- 8 -

Players A and B are subject to a significant latency restriction. In addition, the transmission of rendering commands 56 from server 51 to clients 50 and 52 can be hindered by bandwidth limitations of the network.

5 Consequently, the present invention provides a technique for overcoming such difficulties.

In the technique of the present invention, every machine that participates in the game runs software referred to herein as an "event server." One of the 10 event servers is known as the "master event server" and is responsible for sequencing all changes to a master database representing the game world. All event servers other than the master event server are considered to be "slave event servers".

15 In the preferred embodiment, the master database is located on the same machine as the master event server. Slave copies of the master database (or portions thereof) are stored on each machine other than the machine in which the master event server is located. By 20 providing one master copy of the database, events originating from different machines can be more easily synchronized. By providing slave copies of the database in every other machine involved in the game, the need to send rendering commands across the network is reduced or 25 eliminated, which reduces bandwidth consumption.

In one embodiment of the present invention, the master database is divided into multiple sections, or "objects", each of which may reside on a different machine. Each section functions as a master database 30 relating to a particular aspect or subset of the game, and a separate master event server is associated with each section.

An event server generally performs the following primary functions:

35 (1) connecting to other event servers;

- 9 -

- (2) routing events to the local database and to other event servers;
- (3) migrating the master event server; and
- (4) reconnecting to a new event server such that 5 events are not lost or duplicated.

Each event server maintains a list of open connections to other event servers, one of which is always tagged as the route to the master (RTM) event server.

10 As will be described in greater detail below, the master event server is migrated to a machine used by the player having the role with the most restrictive latency requirement. In the embodiment described above in which there are multiple master event servers corresponding to 15 multiple sections of the master database, each master event server can be independently migrated to another machine, as appropriate. Further, all machines participating in the game which do not have the master event server are then able to reconnect to the machine 20 which has the master event server or to any other machine in order to improve network latency.

Figure 5A illustrates a technique according to the present invention for implementing a multi-user game over the Internet. In Figure 5A, the game is played by two 25 players, players A and B, operating computer systems 60 and 62, respectively. It is noted, however, that the present invention is equally applicable to a game played simultaneously by more than two players. Initially, the master database 71 and the master event server 66 are 30 located in a central server computer system 61. Located in computer system 60 (used by player A) are a slave event queue 65 and a slave copy 68 of master database 71. Similarly, located in computer system 62 (used by player B) is a slave event server 67 and a slave copy 72 of 35 master database 71. The display 74 of computer system 62

- 10 -

is updated by rendering commands 73 from the slave copy 72 of master database 71, rather than from the master database 71 itself, thereby reducing bandwidth consumption on the network. Similarly, the display 70 5 associated with computer system 60 is updated by rendering commands 69 from slave copy 68 of master database 71, rather than directly from master database 71.

Events 64 from computer system 62 are received by 10 slave event server 67 and then passed along to the master event server 66, unless the events 64 are purely local events for updating only the slave copy 72 of the database. Purely local events might be, for example, events which set display options or other functions that 15 are only relevant to computer system 62. Similarly, events 63 originating from computer system 60 are received first by slave event server 65 and then passed along to master event server 66, unless those events are purely local to computer system 60. Events received by 20 the master event server 66 are used to update the master database 71 and are passed along to all slave event servers as master events 75.

Referring now to Figure 7, a routine is illustrated by which by an event server which is not yet 25 involved in the game connects to another event server that is already part of the game. An event server which is not yet part of the game is, by definition, a slave event server. Referring to Figure 7, the slave event server first connects to any other event server, X, which 30 is known to be a part of the game (step 701). The knowledge of which event servers are already part of the game, and their logical addresses, can be provided to the connecting event server in any of a number of possible ways. The methods for providing such knowledge are not 35 germane to the present invention. Next, the slave event

- 11 -

server which is joining the game initializes its connection list to include X (step 702). X is then tagged as the RTM (route to master) in the connection list (step 703).

5 According to the present invention, a determination is made of which computer system participating in the game is used by the player having the role which requires the least latency. Once this is determined, the master event server 66 is migrated to the  
10 machine used by that player. Referring now to Figure 5B, assume that it has been determined that player B (using computer system 62) has the role in the game which requires the least latency. Consequently, master event server 66 is migrated to computer system 62. More  
15 specifically, the status of event server 66 is changed from master event server to slave event server, and the status of event server 67 is changed from slave event server to master event server. Similarly, the master database 71 is migrated to computer system 62. The  
20 status of slave event server 65 associated with Player A remains slave event server. Events 75 output by (newly designated) master event server 67 are then considered to be master events, rather than slave events.

As mentioned above, the present invention is  
25 equally applicable in a situation in which many players using many computers are involved in the game using the Internet as a communication medium. The logical configuration of computers involved in the game in that case would likely have the form of a tree hierarchy, as  
30 illustrated in Figure 3. Figure 3 illustrates the logical configuration of several computer systems 20 through 26. Computer system 20 initially contains the master event server and the master database and is therefore shown as the root node of the tree hierarchy.

- 12 -

In a game situation involving two or more players, a given computer may not be directly connected to the computer having the master event server; such is the case for computers 23 through 26 in Figure 3. Consequently, 5 certain computers, such as computers 21, 22 and 25, will be required to forward events received from other computers. The present invention provides a technique for handling such forwarding, which is discussed below in connection with Figures 8A and 8B.

10 The present invention includes steps by which the slave event server sends slave events to the master event server over the network, and by which the master event server sends (multi-casts) master events over the network to every slave event server in the game. The master 15 event server has the responsibility of ensuring that slave events originating from any given machine are processed in the order in which they were sent. This is a concern, since it is possible for two slave events to arrive out of order when the master event server migrates 20 or when a slave event server reconnects to a different event server (e.g., after migration of the master event server).

Figure 8A illustrates a routine used by a slave event server for opening connections to other event 25 servers and for routing events. Initially, the slave event server accepts any new connections and adds them to its connection list (step 801). Next, the slave event server inputs an event received either from a user or from one of its open connections (step 802). If (in step 30 803) the event is a local event (i.e., if the event is intended only for use on the local machine), then the slave event server sends the event only to its local database (step 804) and the routine repeats from the beginning. If the event is not a local event and the 35 event is a slave event (step 805) which originated on the

- 13 -

local machine (step 806), then the event is assigned the latest event number for the local machine (step 807) and is routed to the RTM (step 808). If, however, the event is a slave event, but the event did not originate on the 5 local machine, then the event is not assigned an event number by the local slave event server, but is simply routed to the RTM (step 808). If the event is neither a local event nor a slave event, then the event is a master event, in which case the event is routed to the local 10 database and to every server connection other than the RTM (step 809).

Figure 8B illustrates a routine used by a master event server for opening connections and routing events to slave event servers. Initially, the master event 15 server accepts any new connections and adds the connections to its connection list (step 821). Next, an event is input to the master event server, either from a user or from an open connection (step 822). If the event is determined to be a local event (step 823), then the 20 event is sent only to the master database. However, if the event is not a local event, then the event is considered to be, and is marked as, a master event (step 825). Hence, when a slave event is received and 25 processed by the master event server, it effectively becomes a master event once it is retransmitted (multi-cast) by the master event server. Once the event is marked as a master event, the master event server checks the event number associated with that event to verify that no earlier events from the sending event server are 30 missing (step 826). It is possible for events to be missing in situations in which a sending slave event server has recently reconnected to a newly assigned master event server. In that situation, two events sent by the slave event server may take different routes to 35 their destination and may therefore arrive out of

- 14 -

sequence. Consequently, once all earlier events have been processed (step 827), the event is sent to the local database and to every event server listed in the connection list of the master event server (step 828).

5 As described above, a key feature of the present invention is that the master event server and the master database are migrated to another machine in order to minimize latency. The authority to initiate migration can be placed with either the master event server or a  
10 slave event server. If the authority to migrate is placed with a slave event server, then a slave event server determines whether it has a latency requirement that is sufficiently stringent to justify migrating the master event server to that machine. A slave server can  
15 make this determination by knowing the topology of the game and the latencies of the participating devices. Accordingly, in such an embodiment, the master event server can broadcast information describing the current network configuration and latencies associated with each  
20 device in the game to each connected slave event server at regular intervals, or whenever a new slave event server connects to the game.

Regardless of which device has the actual authority to approve of migration of the master event  
25 server, the migration process can be initiated by a request or command transmitted over the network by a slave event server. The request or command is a special event known as a "slave migration event." The master event server initiates the migration process by sending a  
30 "master migration event". When a slave event server receives a slave migration event from another slave event server, it adds its identifier to the end of the event message. Hence, the slave migration event gets annotated with the complete path it follows to reach the master  
35 event server. The master migration event sent in reply

- 15 -

also contains the same path information. Accordingly, interceding slave event servers can correctly route the master migration event to the slave event server that transmitted the slave migration event and update their 5 respective RTM information.

Figure 9A illustrates the migration routine performed by the master event server according to one embodiment. Initially, the master event server inputs a slave migration event (step 901). The master event 10 server then determines whether to relinquish its status as the master event server (step 902). (As already mentioned, this decision might be made by a slave event server in another embodiment.) If it is appropriate to relinquish its status as the master event server, then 15 the master event server sends a master migration event along the path which was designated by the received slave migration event, and marks that connection as the RTM (step 903). Next, the master event server changes its status to slave event server (step 904). If the master 20 event server determines that it is not appropriate to change its status as master event server (step 902), then the slave migration event is ignored and the routine ends.

Figure 9B illustrates the migration routine used 25 by a slave event server according to one embodiment. Initially, a local slave event server inputs a master migration event that was sent in reply to a slave migration event sent by one of the slave event servers (step 921). Next, the local slave event server 30 determines which event server, X, should next receive the master migration event (step 922). This determination is made on the basis of the path designated by the master migration event. Next, the local slave event server determines whether X is a valid server (step 923). If 35 not, the local slave event server changes its own status

- 16 -

to master event server, despite the fact that it was not the slave server which had requested to become the master (step 927). If X is a valid event server, then the local slave event server determines if X is still connected 5 (step 924) . If not, the local slave event server changes its own status to master event server (step 927) .

Therefore, if a given slave event server requests to become the master event server and subsequently becomes disconnected from the network before the master migration 10 event is received, the master event server will be migrated by default to the last slave event server along the path toward the disconnected slave event server.

Note that there is no master event server until the master migration event reaches its destination. Hence, 15 this step (927) ensures that some event server will always assume the status of master event server if a connection is lost. Assuming X is a valid event server and is still connected, the receiving slave event server passes the master migration event to the next slave event 20 server along the designated path (step 925) .

Accordingly, the connection through which the master migration event is passed is designated as the RTM by the local slave event server.

Figure 9C illustrates a routine by which a master 25 event server can determine whether to relinquish its status as master event server. Upon receiving a slave migration event, the master event server identifies the role in the game that is associated with the slave event server that sent the slave migration event (step 941) .

30 If that role requires the lowest latency of any role in the game, then it is appropriate to migrate the master event server. Accordingly, the master event server relinquishes its status as master event server (step 943) . Otherwise, the master event server retains its 35 status as master (step 944) .

- 17 -

Referring again to Figure 5B, once the master event server has been migrated to computer system 62, it may not be desirable from a latency standpoint (or for other reasons) to require events originating from 5 computer system 60 to pass through the central server computer system 61. For example, once the master event server has been migrated to a different machine, it may be appropriate for one slave event server to reconnect to another slave event server to reduce its latency.

10 Referring to both Figure 5B and Figure 6, note that slave event server 65 on computer 60 was initially connected to event server 66 on computer 61 (Figure 5B). Event server 66 was initially the master event server. However, once event server 67 in computer 62 becomes the 15 master event server, it becomes more efficient for slave event server 65 to connect directly to event server 67 (Figure 6). Consequently, the present invention allows any given slave event server in the game to alter its connections in order to improve network latency. For 20 example, slave event server 65 can establish a new connection directly to the newly designated master event server 67 and drop its existing connection to the former master event server 66, as illustrated in Figure 6.

Figure 10 illustrates a routine for performing 25 such reconnection. In one embodiment, the reconnection routine is initiated in response to a message from the former master event server, indicating that the receiving event server should reconnect to the new master event server. In Figure 10, upon receiving the signal to 30 reconnect, a slave event server opens a new connection to the newly-designated master event server (or to another slave event server, if appropriate), and adds that connection to the connection list (step 1001). Next, the slave event server waits until the event streams from the 35 new connection and the old connection overlap (step

- 18 -

1002). In other words, the slave event server waits until it sees a sequence of events that is the same in the streams coming from both the old connection and the new connection. When the streams overlap, the event 5 server closes the old connection and deletes the old connection from its connection list (step 1003). In addition, the slave event server maintains knowledge of which events have been processed in order to avoid duplication of events (step 1004).

10 Thus, a technique for reducing latency when synchronizing access to a multi-user database over a network has been described. Although the present invention has been described with reference to specific exemplary embodiments, it will be evident that various 15 modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention as set forth in the claims. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive 20 sense.

- 19 -

CLAIMS

What is claimed is:

1. In a network having a plurality of nodes including a first node and a second node, the first node 5 including a first event server for sequencing commands to a database stored in one of the nodes, the second node including a second event server for sequencing commands to the database, the first event server and the second event server each having a status, a method of reducing 10 network latency in accessing the database, the method comprising the steps of:

(a) changing the status of the first event server from master event server to slave event server based on a network latency criterion; and

15 (b) assigning the status of master event server to the second event server based on the network latency criterion.

2. The method of claim 1, wherein the step of changing the status of the first event server from master 20 event server to slave event server based on a network latency criterion comprises the step of changing the status of the first event server from master event server to slave event server if the second node has a more restrictive network latency requirement than the first 25 node.

3. In a network having a plurality of nodes, the plurality of nodes including a local node and a plurality of remote nodes, a first one of the remote nodes having a master database and a master event server 30 for sequencing commands for the master database, a method of reducing network latency in accessing the master database, the method comprising the steps of:

- 20 -

- (a) migrating the master database to the local node;
- (b) migrating the master event server to the local node; and
- 5 (c) inputting a slave event to the master event server migrated to the local node for execution by the local node against the master database migrated to the local node.

4. The method of claim 3, wherein the local node 10 has a slave event server, and wherein the migrating step comprises the steps of:

- (a) changing the status of the master event server to slave event server; and
- (b) changing the status of the slave event 15 server to master event server.

5. The method of claim 3, wherein the first one of the remote nodes includes a first slave event server, the method further comprising the steps of:

- (a) inputting to the first one of the remote nodes a slave event received from a second one of the remote nodes;
- (b) forwarding the slave event from the first one of the remote nodes to the master event server migrated to the local node for execution by the local node against the master database;
- 25 (c) inputting to the first slave event server an event received from the master event server; and
- (d) forwarding the event received from the master event server migrated to the local node from the first one of the remote nodes to the second one of the remote nodes.

- 21 -

6. The method of claim 4, wherein the second one of the remote nodes includes a second slave event server, the method further comprising the steps of:

5 (a) disconnecting the second slave event server from the first slave event server; and  
(b) connecting the second slave event server directly to the master event server migrated to the local node.

7. The method of claim 4, wherein the second one 10 of the remote nodes includes a second slave event server, the method further comprising the steps of:

15 (a) disconnecting the second slave event server from the first one of the remote nodes; and  
(b) connecting the second slave event server directly to a third one of the remote nodes.

8. The method of claim 3, wherein the step of migrating the master event server comprises the step of migrating the master event server to the local node based on a network latency criterion.

20 9. The method of claim 8, wherein the step of migrating the master database comprises the step of migrating the master database to the local node based on a network latency criterion.

25 10. The method of claim 3, further comprising the step of selecting the local node as the node to which the master event server is to be migrated based on a network latency criterion.

30 11. The method of claim 3, wherein the local node is the node in the network having the most restrictive network latency requirement of the nodes in the network.

- 22 -

12. In a network including a first node in communication with a plurality of other nodes, a method of implementing a machine-executable game having a plurality of players, the game being generated based on a 5 master database and a master event server, the master event server for receiving and storing a plurality of slave events for updating the master database, the master event server and the master database being initially stored in the first node, each player using one of said 10 other nodes to perform one of a plurality of roles in the game, the method comprising the steps of:

15 (a) determining one of said other nodes to which the master event server will be migrated based on a network latency criterion associated with at least one of the roles; and  
(b) migrating the master event server to said one of said other nodes.

13. The method of claim 12, wherein the migrating step comprises the step of changing the status of the 20 master event server to that of slave event server.

14. The method of claim 13, wherein said one of said other nodes to which the master event server is migrated has a slave event server, and wherein the migrating step further comprises the step of changing the 25 status of the slave event server to that of master event server.

15. The method of claim 12, wherein a first one of said other nodes includes a first slave event server, and wherein a second one of said other nodes includes a 30 second slave event server, the method further comprising the steps of:

- 23 -

(a) inputting to the first one of said other nodes a slave event received from the second one of said other nodes;

5 (b) forwarding the slave event from the first one of said other nodes to the migrated master event server migrated to said one of said other nodes;

10 (c) inputting to the first slave event server a master event received from the master event server migrated to said one of said other nodes; and

15 (d) forwarding the master event received from the master event server migrated to said one of said other nodes to the second slave event server.

16. The method of claim 12, wherein the network includes a communication link between the first slave event server and the second slave event server, the method further comprising the steps of:

20 (a) disconnecting the communication link between the first slave event server and the second slave event server; and

25 (b) establishing a communication link between the second slave event server and the migrated master event server, such that requests initiated from the second one of said other nodes are transmitted to the master event server migrated to said one of said other nodes without passing through the first one of said other nodes.

30 17. The method of claim 12, wherein a third one of said other nodes includes a third slave event server, wherein the network includes a communication link between the first slave event server and the second slave event

- 24 -

server, and wherein the method further comprises the steps of:

18. The method of claim 12, further comprising  
the step of using the third node to forward a slave  
event, the slave event received from whichever one of the  
15 first node and the second node is used by the player not  
having the role requiring the lowest level of latency, to  
the other one of the first node and the second node.

19. The method of claim 12, wherein the network is a wide-area network.

20. The method of claim 12, wherein the step of determining comprises the step of determining said one of said other nodes to which the master event server should be migrated based on the role of the player using said one of said other nodes.

25           21. The method of claim 12, further comprising  
the step of migrating at least a portion of the master  
database based on the network latency criterion.

22. The method of claim 12, further comprising  
the step of migrating a copy of at least a portion of the  
30 master database to each of said other nodes.

- 25 -

23. The method of claim 12, wherein the network latency criterion is a target latency for a given one of the roles.

24. The method of claim 12, wherein the network 5 latency criterion is a maximum allowable latency for a given one of the roles.

25. In a network in which a first node and a second node are coupled to a third node, a method of reducing network latency during execution of computer 10 program code embodying a multiple-player game, a first player of the multi-player game participating in the game using the first node and a second player of the multi-player game participating in the game using the second node, the game being characterized by a master event 15 server receiving and storing user inputs and a database representing a game environment in response to the user inputs, the master event server and the database being initially stored on the third node, the method comprising the steps of:

20 (a) determining which of the players has a role in the game requiring a lowest level of latency;

25 (b) identifying which one of the first node and the second node is used by the player having the role requiring the lowest level of latency; and

30 (c) migrating the master event server to said one of the first node and the second node that is used by the player having the role requiring the lowest level of latency.

- 26 -

26. The method of claim 25, wherein the migrating step comprises the step of changing the status of the master event server to that of slave event server.

27. The method of claim 25, wherein said one of 5 the first node and the second node that is used by the player having the role requiring the lowest level of latency has a slave event server, and wherein the migrating step further comprises the step of changing the status of the slave event server to that of master event 10 server.

28. The method of claim 25, further comprising the step of using the third node to forward a slave event to said one of the first node and the second node that is used by the player having the role requiring the 15 lowest level of latency.

29. The method of claim 25, further comprising the step of migrating the database to said one of the first node and the second node that is used by the player having the role requiring the lowest level of latency.

20 30. The method of claim 25, further comprising the step of placing a copy of the database in each of the first node and the second node, such that each said copy is updated from the contents of the master event server migrated to said one of the first node and the second 25 node that is used by the player having the role requiring the lowest level of latency.

- 27 -

31. An apparatus for reducing latency in a network having a plurality of processing systems, the apparatus comprising:

5 (a) a first processing system generating a first portion of a multi-player game, the first processing system having stored therein an event server, the event server receiving and storing user inputs for controlling the game, the first processing system further having stored therein a 10 database representing a game environment responsive to the user inputs;

15 (b) a second processing system coupled to the first processing system, the second processing system generating a second portion of the multi-player game operable by a first player of the multi-player game, the first player performing a first role in the multi-player game, the first role having a first latency requirement;

20 (c) a third processing system coupled to the first processing system, the third processing system generating a third portion of the multi-player game operable by a second player of the multi-player game, the second player performing a second role in the multi-player game, the second role having a second latency requirement; and

25 (d) the first processing system evaluating the first latency requirement and the second latency requirement and, in response, migrating the event server to either the second memory or the third memory according to a relationship 30 between the first latency requirement and the second latency requirement.

32. The apparatus of claim 31, wherein the database is migrated to either the second memory or the

- 28 -

third memory according to the relationship between the first latency requirement and the second latency requirement.

33. The apparatus of claim 31, further comprising  
5 means for migrating a copy of the database to both the first processing system and the second processing system.

34. The apparatus of claim 31, wherein the latency requirement is a target latency for a given one of the roles.

10 35. The apparatus of claim 31, wherein the latency requirement is a maximum allowable latency for a given one of the roles.

36. The apparatus of claim 31, wherein the first processing system forwards a slave event received from  
15 whichever of the second processing system and the third processing system does not have the master event server to whichever of the second processing system and the third processing system has the master event server.

37. The apparatus of claim 31, wherein the first  
20 processing system is further configured:

(a) to determine when the master event server has been migrated to the third processing system, and

25 (b) to transmit the slave event directly to the master event server in the third processing system, such that the slave event does not pass through the second processing system.

38. In a network having a plurality of nodes, the plurality of nodes including a local node and a plurality

- 29 -

of remote nodes, at least one of the remote nodes having a database and a master event server for sequencing commands for the database, an apparatus for reducing network latency in accessing the database, the apparatus 5 comprising:

- (a) means for migrating a master copy of the database to the local node;
- (b) means for migrating the master event server to the local node; and
- 10 (c) means for forwarding a slave event from at least one of the remote nodes to the master event server migrated to the local node for execution by the local node against the database migrated to the local node.

15 39. The apparatus of claim 38, wherein the local node has a slave event server, and wherein the migrating step comprises the steps of:

- (a) changing the status of the master event server to that of slave event server; and
- 20 (b) changing the status of the slave event server to that of master event server.

40. The apparatus of claim 38, wherein a first one of the remote nodes includes a first slave event server, the apparatus further comprising:

- 25 (a) means for inputting to the first one of the remote nodes a slave event received from a second one of the remote nodes;
- (b) means for forwarding the slave event from the first one of the remote nodes to the master event server migrated to the local node for execution by the local node against the master database;

- 30 -

(c) means for inputting to the first slave event server a master event received from the master event server; and

5 (d) means for forwarding the master event received from the master event server migrated to the local node to the second one of the remote nodes.

10 41. The apparatus of claim 39, wherein the second one of the remote nodes includes a second slave event server, the method further comprising the steps of:

(a) means for disconnecting the second slave event server from the first slave event server; and

15 (b) means for connecting the second slave event server directly to the master event server migrated to the local node.

42. The apparatus of claim 39, wherein the second one of the remote nodes includes a second slave event server, the method further comprising the steps of:

20 (a) means for disconnecting the second slave event server from the first slave event server; and

25 (b) means for connecting the second slave event server directly to a third one of the remote nodes, such that slave events transmitted by the second event server and destined for the master event server do not pass through the first one of the remote nodes.

43. The apparatus of claim 38, wherein the means 30 for migrating the event server comprises means for migrating the master event server to the local node based on a network latency criterion.

- 31 -

44. The apparatus of claim 43, wherein the network latency requirement is a target latency value.

45. The apparatus of claim 43, wherein the network latency requirement is a maximum allowable 5 latency for the local node.

46. The apparatus of claim 43, wherein the means for migrating the master copy of the database comprises means for migrating the master copy of the database to the local node based on a network latency criterion.

10 47. The apparatus of claim 38, further comprising means for selecting the local node as the node to which the master event server is to be migrated based on a network latency criterion.

15 48. The apparatus of claim 38, wherein the local node is the node in the network having the most restrictive network latency requirement of the nodes the network.

49. The apparatus of claim 38, further comprising:

20 (a) means for evaluating a bandwidth criterion associated with at least one of the nodes in the network;

25 (b) wherein the means for migrating a master copy of the database includes means for migrating the master copy of the database to the local node based on an output of the means for evaluating.

50. A machine-readable program storage medium tangibly embodying a program of instructions, the instructions executable by a server computer to perform

- 32 -

method steps for implementing a game using a computer network, the computer network including the server computer and a plurality of client computers, the game having a plurality of players, each player having a 5 different role in the game by using a different one of said client computers, the game characterized by a master event server storing user inputs for updating a master database, the method steps comprising the steps of:

- 10 (a) providing a latency criterion for at least one of the roles in the game;
- (b) determining, for at least one of the roles having a latency criterion, which of the client computers in the network is associated with said role;
- 15 (c) identifying a first one of the client computers that is used by a player having the role with the most restrictive latency criterion of all the roles; and
- 20 (d) migrating the master event server from the server computer to the first one of the client computers.

51. The program storage medium of claim 50 wherein the local node has a slave event server, and wherein the migrating step comprises the steps of:

- 25 (a) changing the status of the master event server to that of slave event server; and
- (b) changing the status of the slave event server to that of master event server.

52. The program storage medium of claim 50, 30 wherein the method steps further comprise the step of using the server computer to forward a slave event received from a second one of the client computers to the

- 33 -

master event server migrated to the first one of the client computers.

53. The program storage medium of claim 50, wherein the method steps further comprise the step of 5 using the server computer to forward a master event received from the master event server migrated to the first one of the client computers to a second one of the client computers.

54. The program storage medium of claim 52, 10 wherein the method steps further comprise the steps of:

(a) providing an indication to the second one of the client computers that the master event server has been migrated to said particular one of the client computers; and

15 (b) providing a direct link between the second one of the client computers and the master event server, such that slave events initiated by the second client computer do not pass through the server computer.

20 55. The program storage medium of claim 50, wherein the method steps further comprise the step of migrating a copy of at least a portion of the master database to at least one of the client computers to which the master event server has not been migrated.

25 56. A method of communicating events in a network including a plurality of nodes, one of the plurality of nodes having a master database and a master event server stored therein, the master event server for sequencing commands for the master database, the method comprising 30 the steps of:

- 34 -

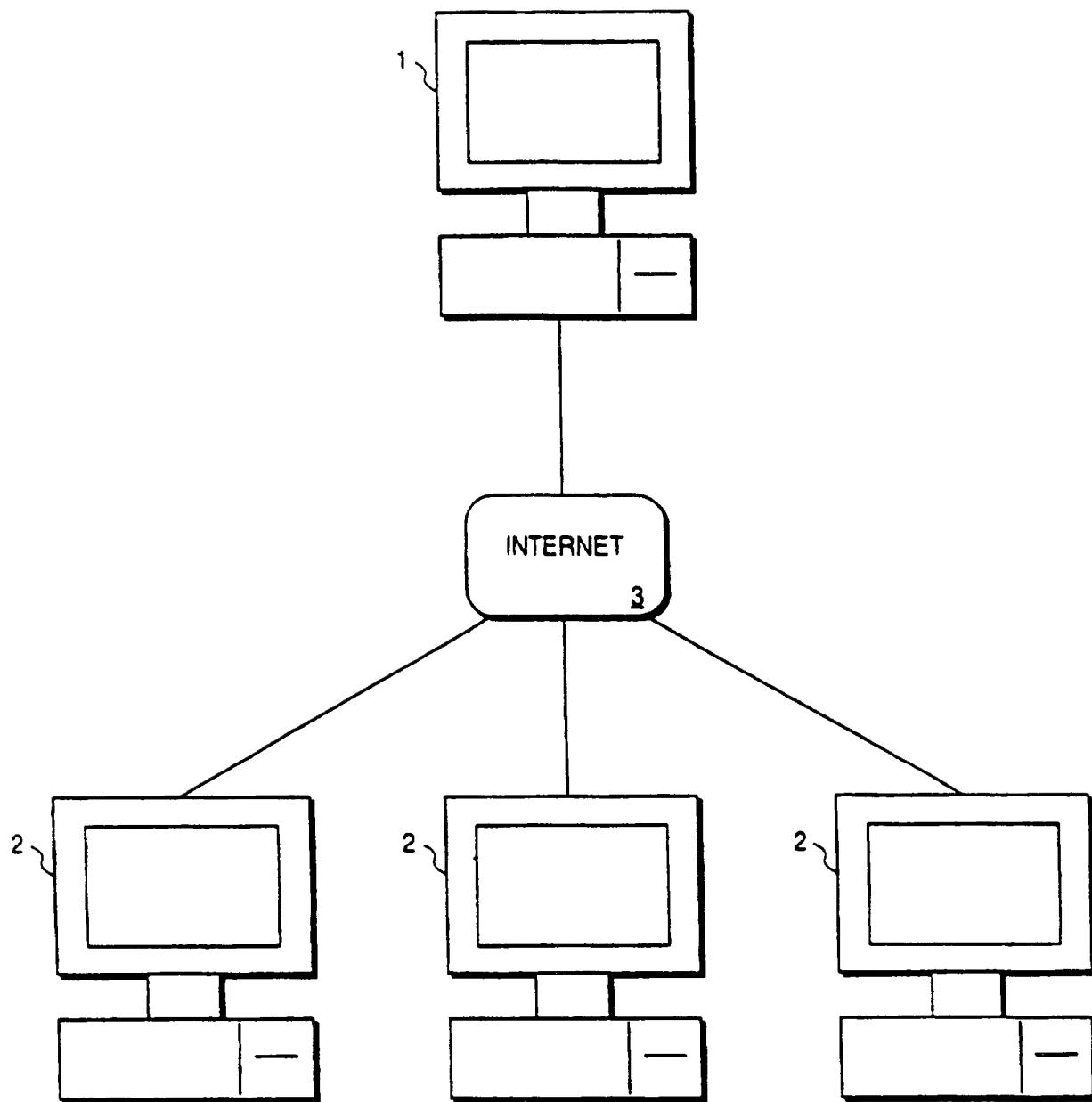
- (a) inputting to a first one of the nodes a slave event received from a second one of the remote nodes;
- 5 (b) determining whether the master event server resides within the first one of the nodes;
- (c) if the master event server does not reside within the first one of the nodes, identifying a third one of the plurality of nodes in which the master event server resides; and
- 10 (d) if the master event server does not reside within the first one of the nodes, forwarding the slave event to the master event server in the third one of the nodes.

57. The method of claim 56, further comprising  
15 the step of establishing a communication link between the second one of the nodes and the third one of the nodes, such that a second slave event initiated by the second one of the nodes is transferred to the third one of the nodes without passing through any other nodes of the  
20 plurality of nodes.

58. The method of claim 56, wherein the master event server has been migrated from the first one of the nodes to the third one of the nodes prior to the inputting step.

- 35 -

59. The method of claim 56, wherein the first one of the nodes includes a first event server, wherein the first event server can have a first status in which the first event server is the master event server, or a 5 second status in which the first event server is a slave event server, the method further comprising the step of switching the status of the first event server from the first status to the second status prior to the inputting step.

**FIG. 1**

2/15

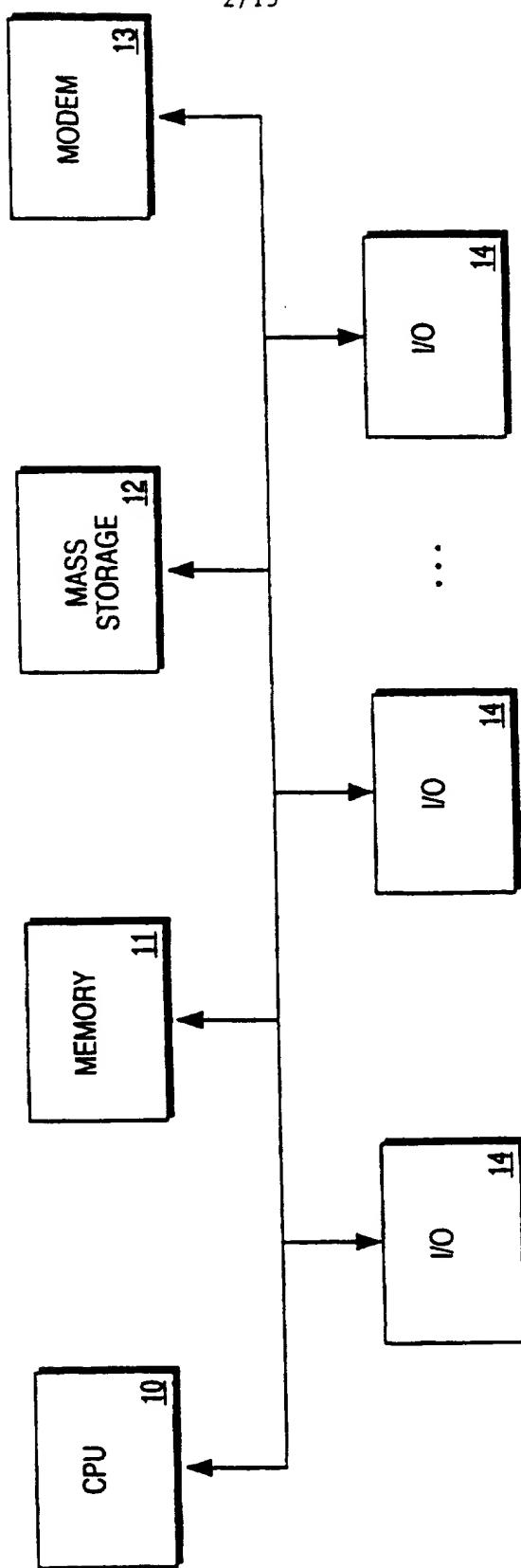
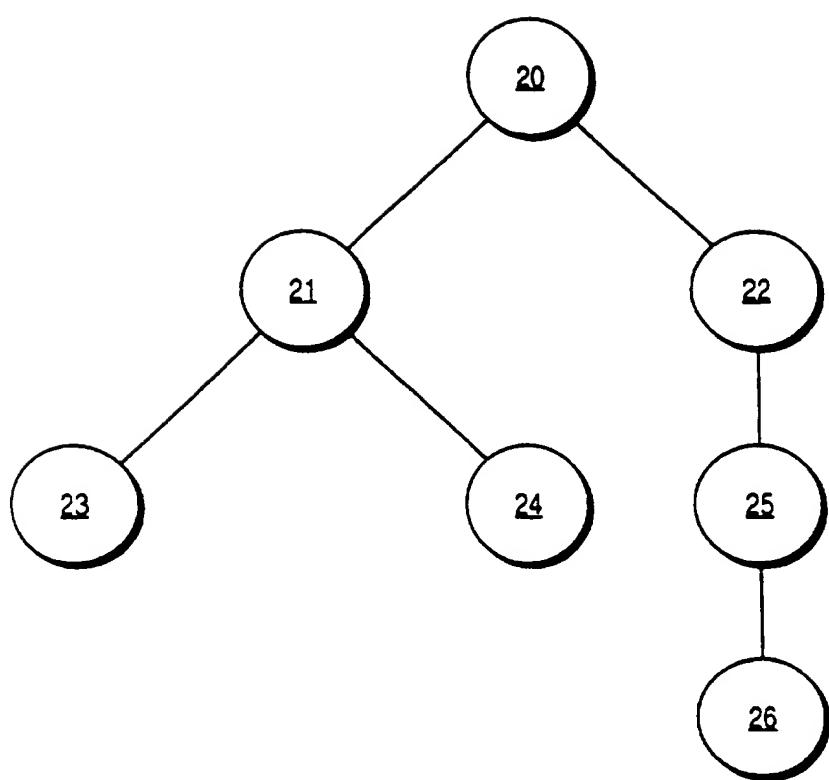
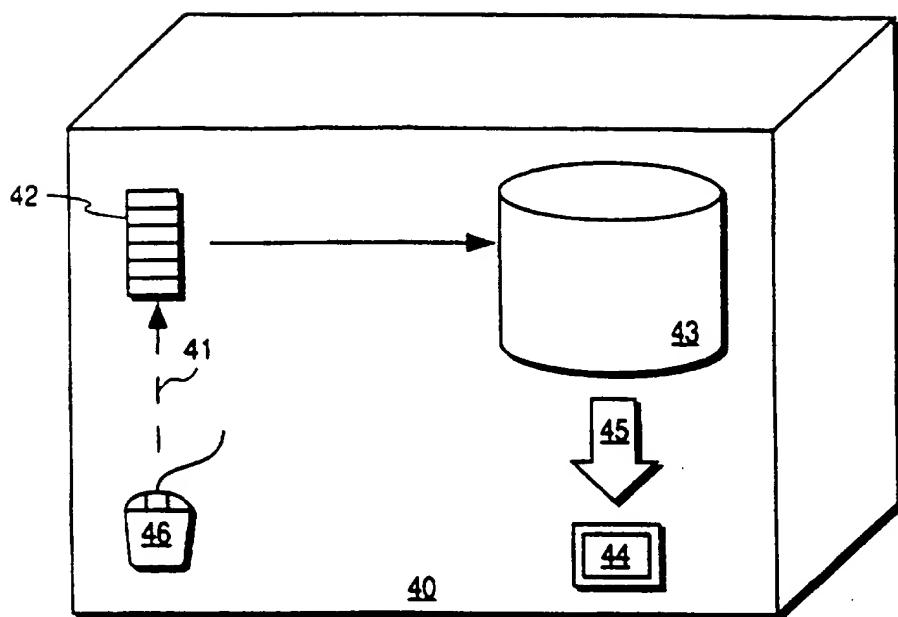


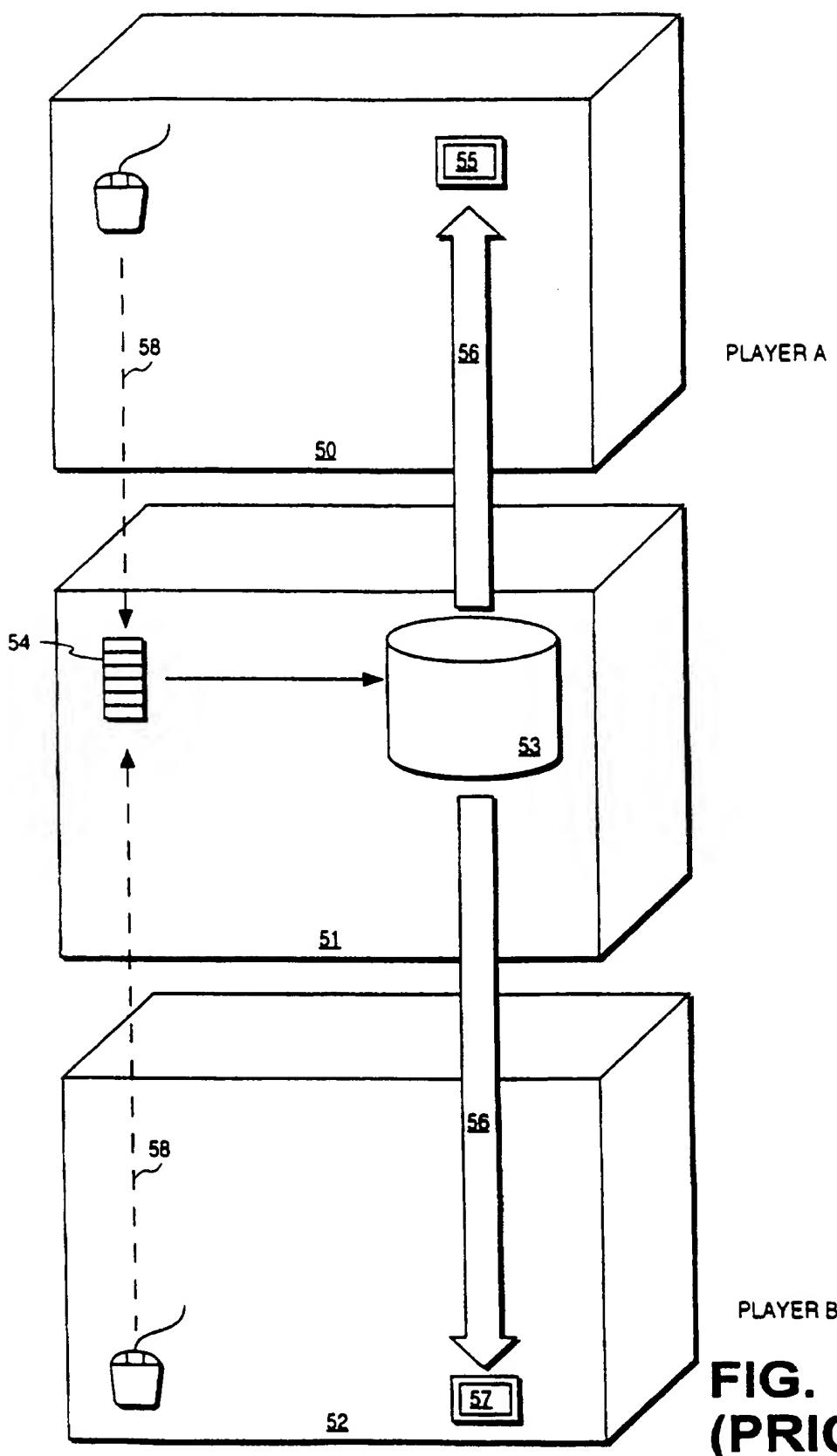
FIG. 2



**FIG. 3**

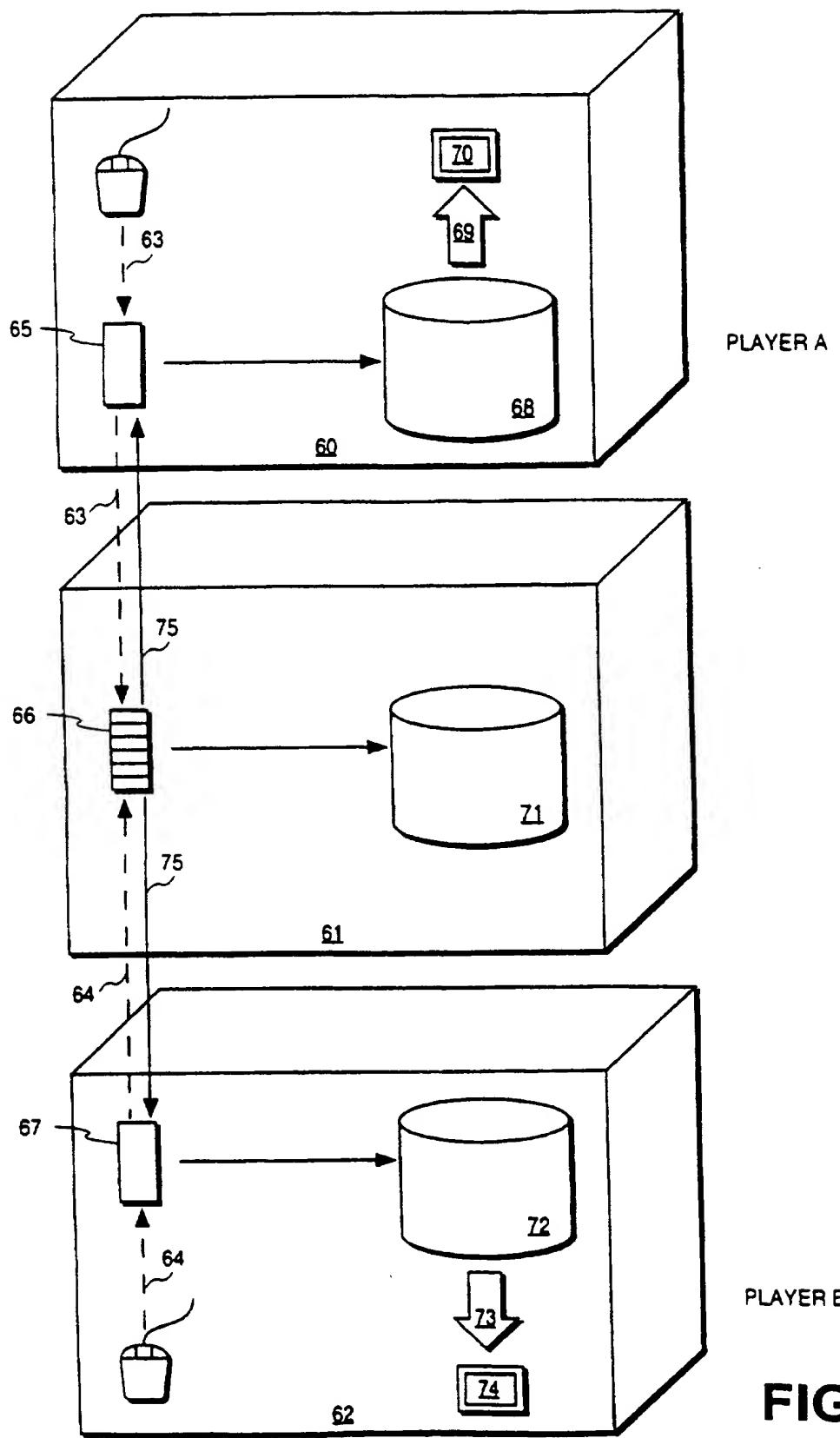


**FIG. 4A  
(PRIOR ART)**

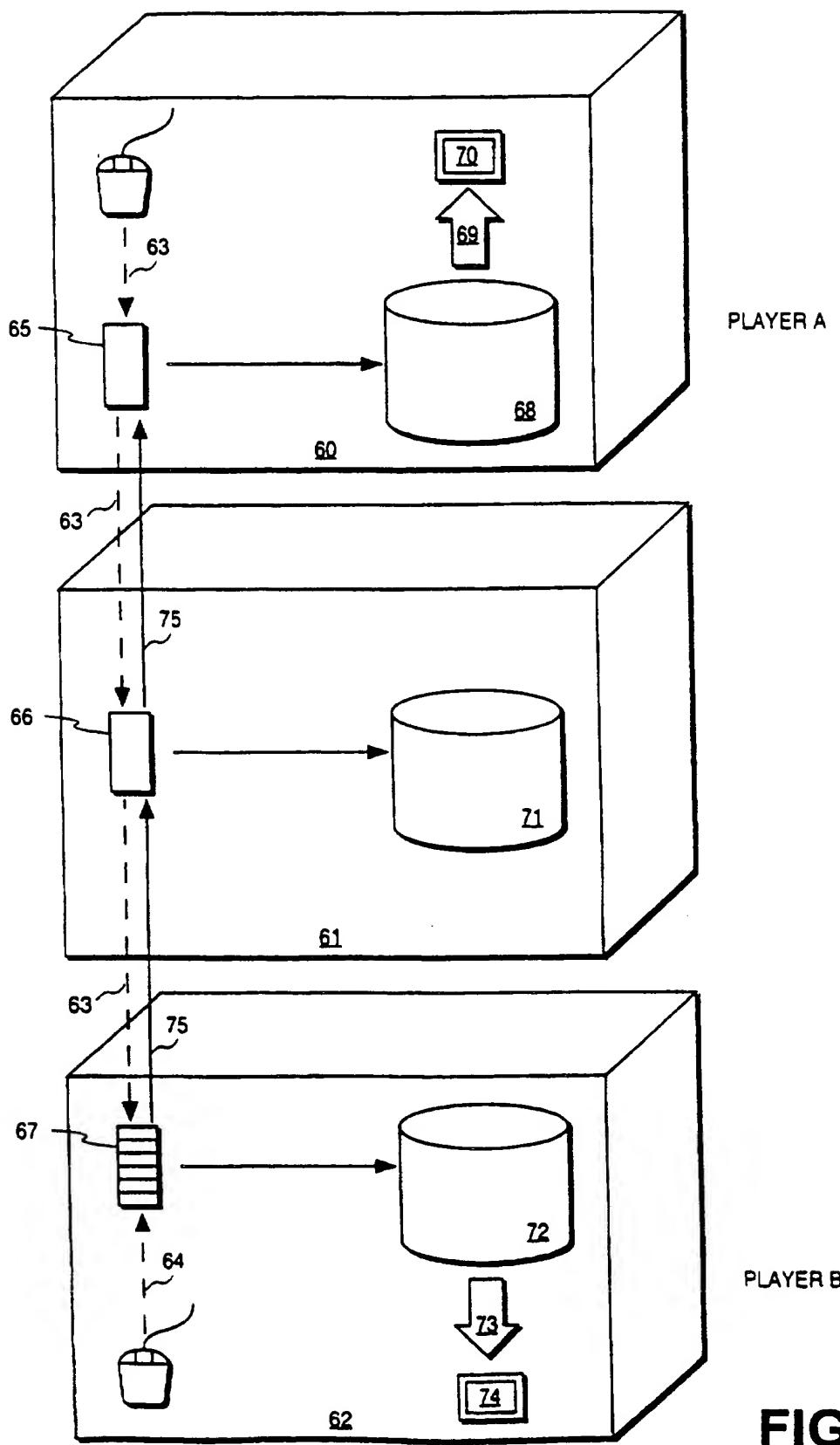


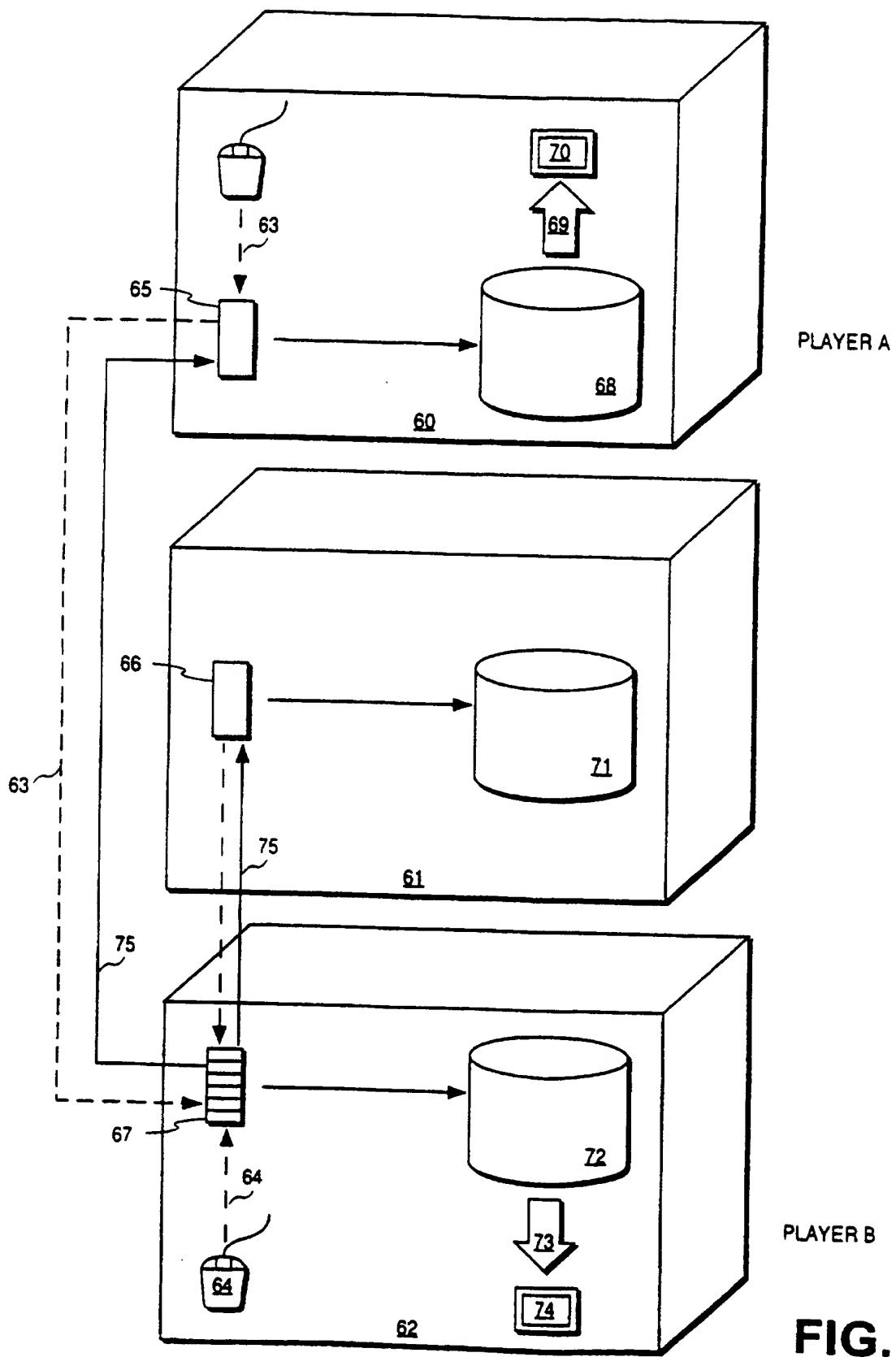
**FIG. 4B  
(PRIOR ART)**

6/15

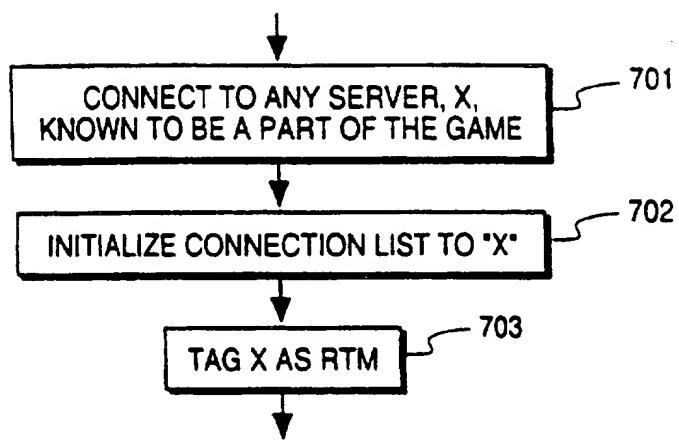
**FIG. 5A**

7/15

**FIG. 5B**

**FIG. 6**

9/15

**FIG. 7**

10/15

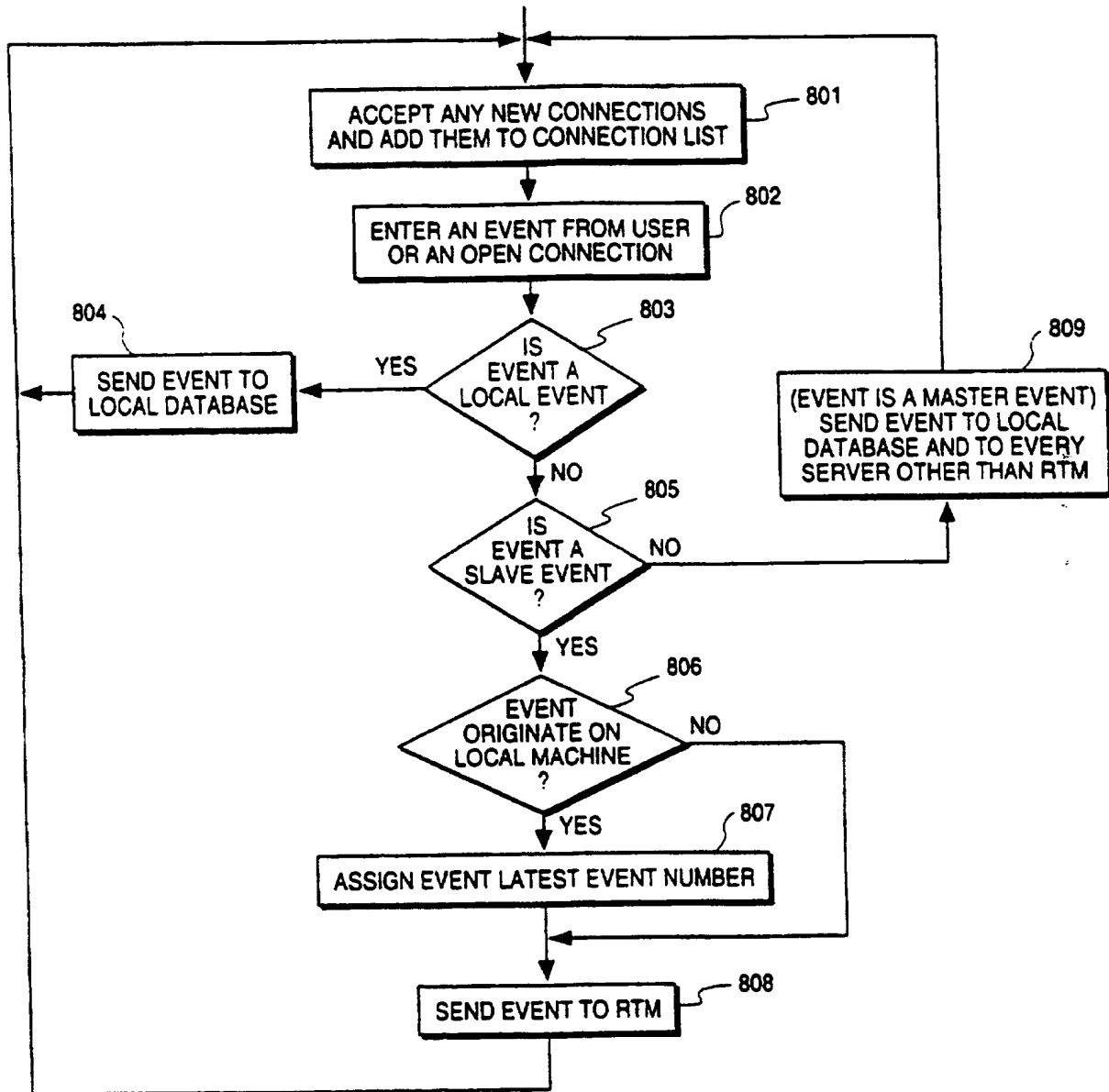


FIG. 8A

11/15

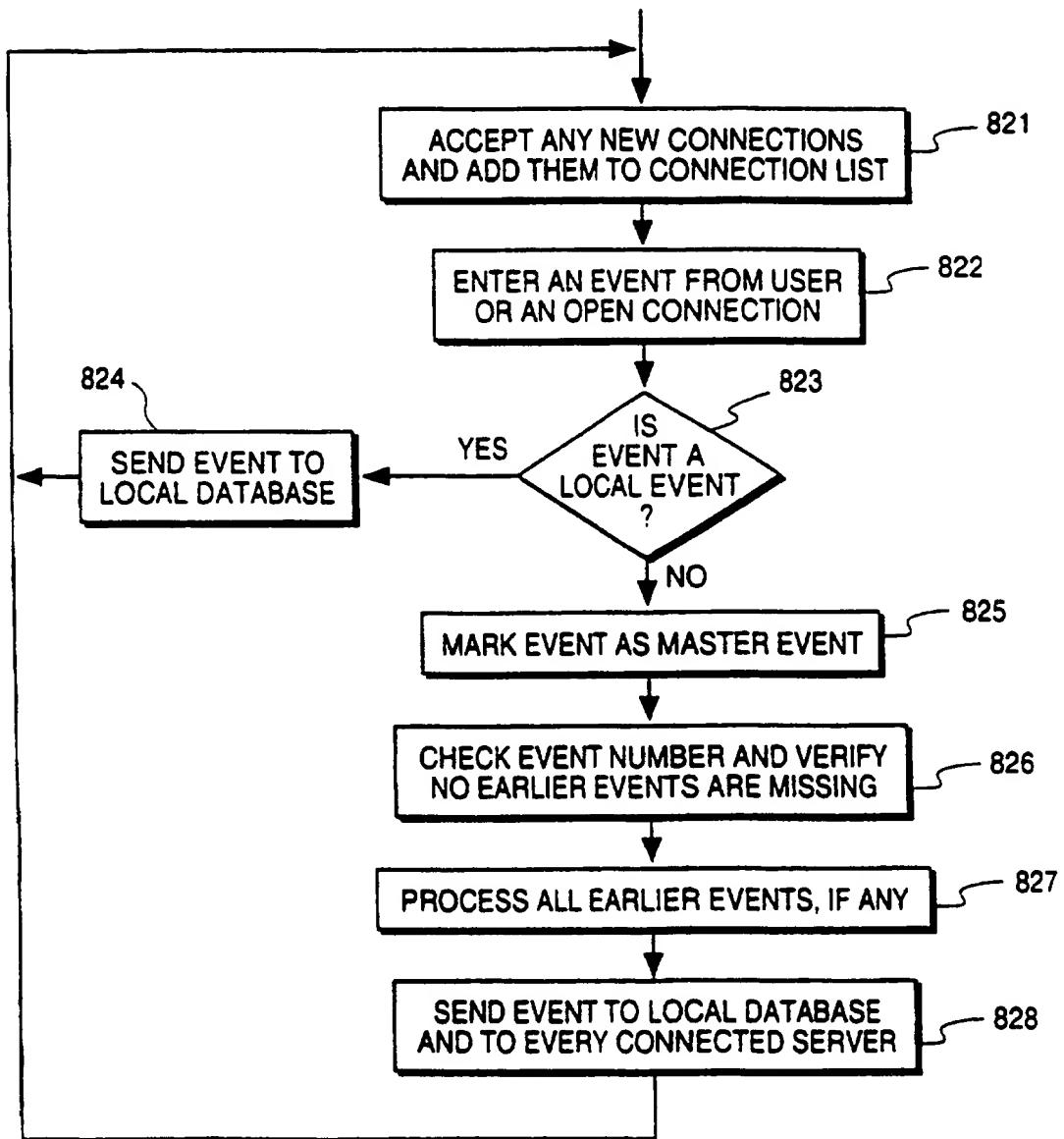
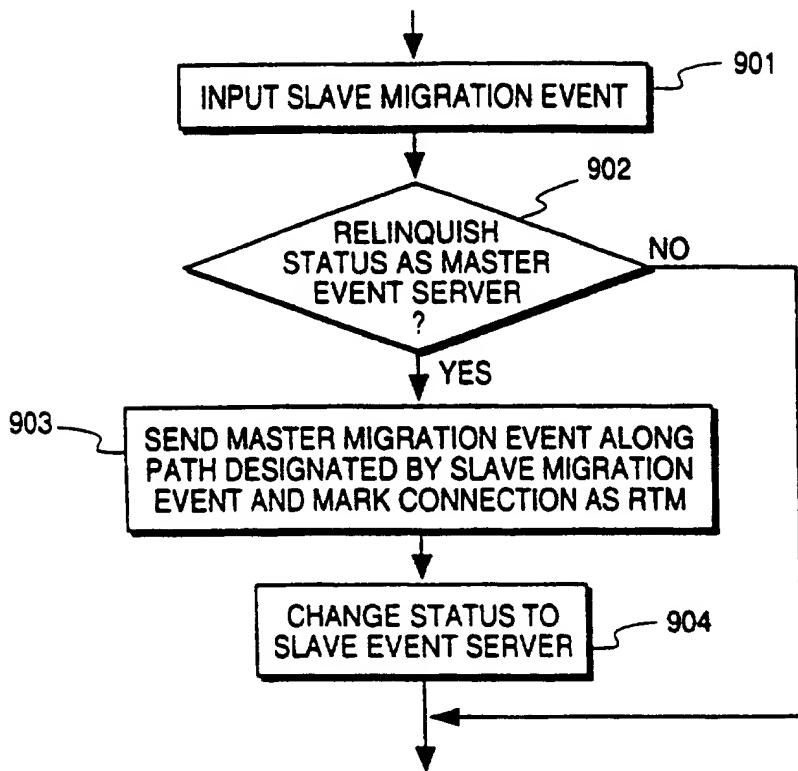


FIG. 8B

12/15

**FIG. 9A**

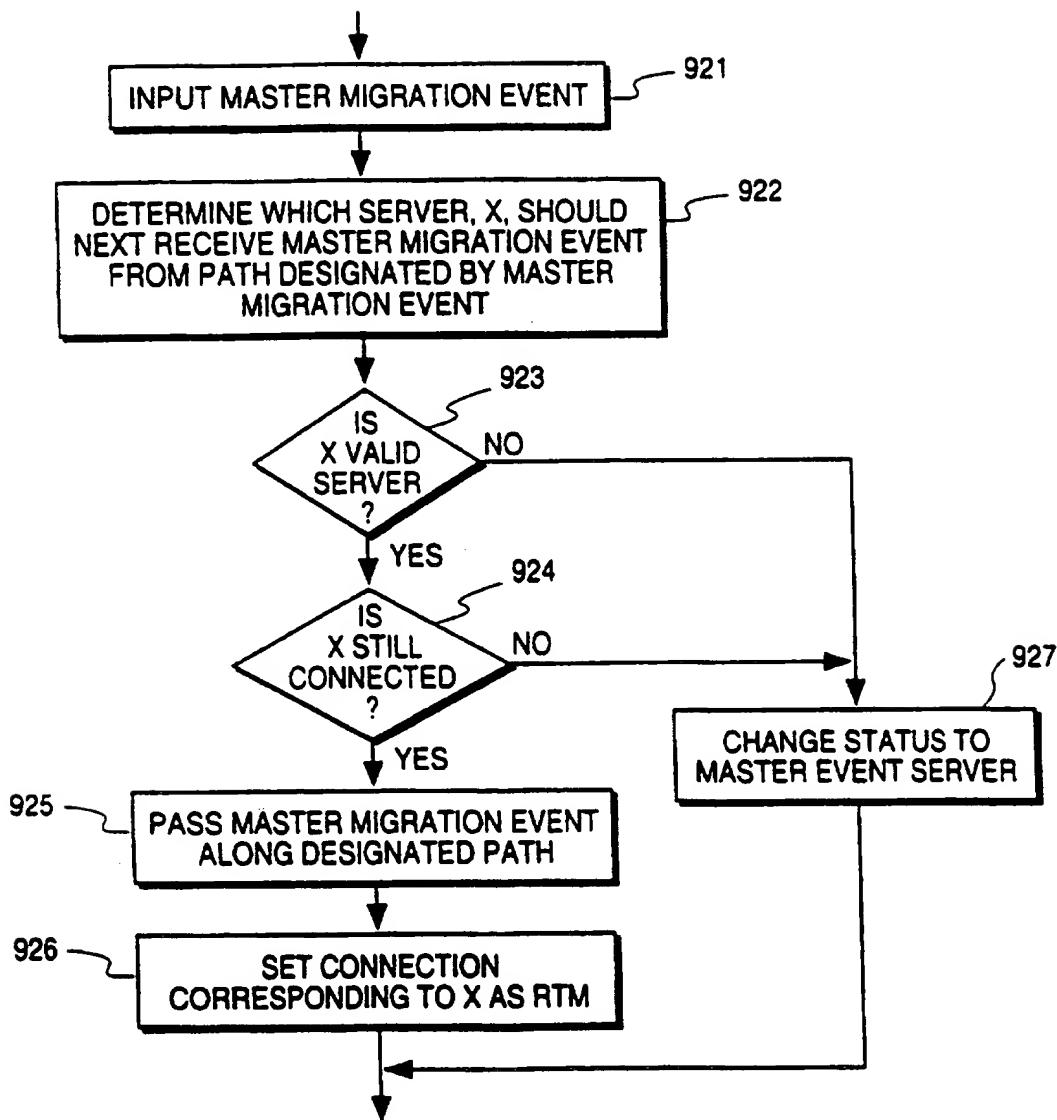
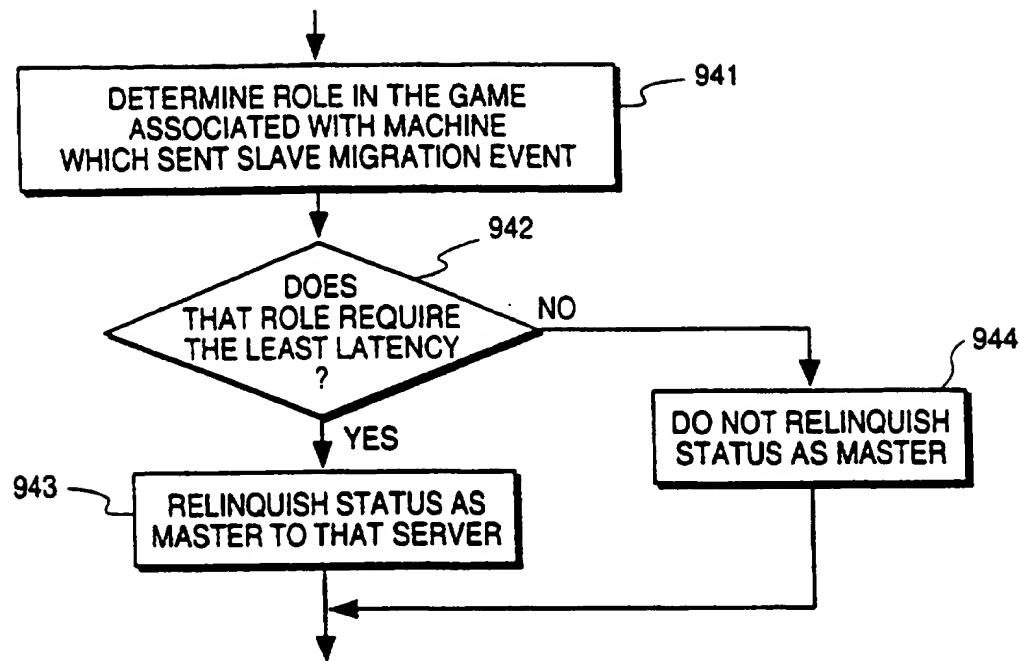
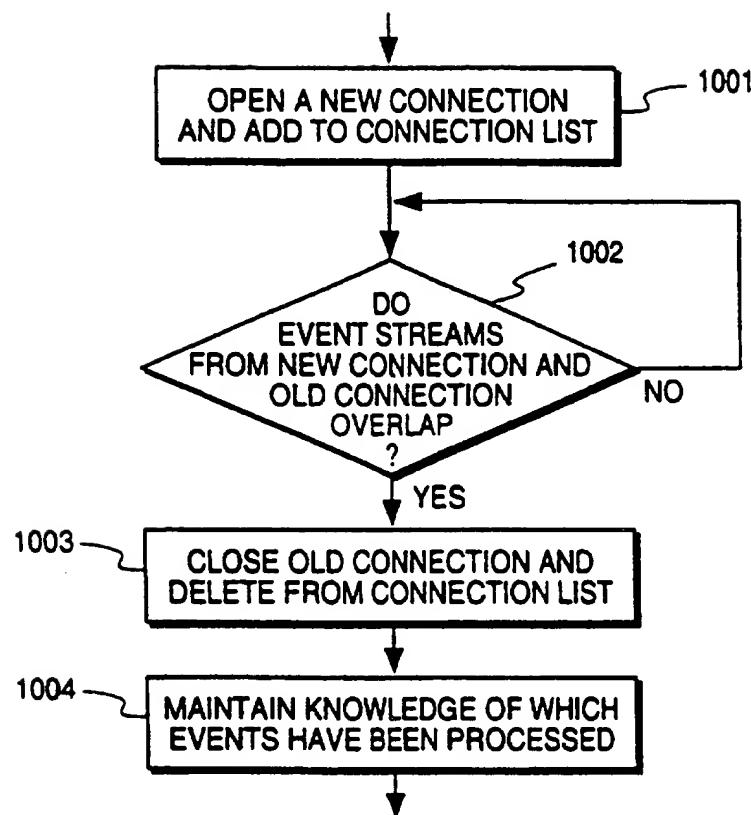


FIG. 9B

**FIG. 9C**

15/15

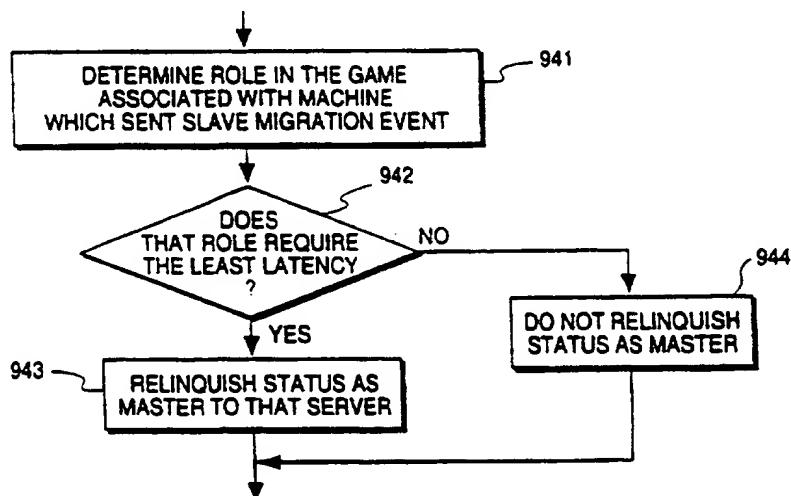
**FIG. 10**



## INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : <b>G06F 15/16</b>		A3	(11) International Publication Number: <b>WO 97/47091</b> (43) International Publication Date: 11 December 1997 (11.12.97)
(21) International Application Number: <b>PCT/US97/09054</b> (22) International Filing Date: 22 May 1997 (22.05.97)		(81) Designated States: AU, CA, JP, KR, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).	
(30) Priority Data: 08/652,803 23 May 1996 (23.05.96) US		Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>	
(71) Applicant: ADOBE SYSTEMS INCORPORATED (US/US); 345 Park Avenue, San Jose, CA 95110 (US).		(88) Date of publication of the international search report: 15 January 1998 (15.01.98)	
(72) Inventors: ROY, H., Scott; 1365 Green Street #304, San Francisco, CA 94109 (US). HARVEY, William, D.; Apartment 101, 432 High Street, Palo Alto, CA 94301 (US).			
(74) Agent: TROESCH, Hans, R.; Fish & Richardson P.C., Suite 100, 2200 Sand Hill Road, Menlo Park, CA 94025 (US).			

(54) Title: REDUCING LATENCY WHEN SYNCHRONIZING ACCESS TO A MULTI-USER DATABASE OVER A NETWORK



## (57) Abstract

A master database represents a world model for the game, and a master event server sequences user inputs, or events, for updating the master database. Slave event servers are responsible for updating their local slave database, sending events from their local node to the master event server, and forwarding events to and from other slave servers. In the method, a determination is made of which player in the game has a role in the game requiring the least latency of any role (941). The master event server and the master database are then migrated to that node by changing the status of that slave event server to master event server (943). Once the master event server is migrated, other slave event servers can establish a direct connection with the new master event server.

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		

## INTERNATIONAL SEARCH REPORT

International application No.

PCT/US97/09054

## A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 15/16  
US CL :395/200.39

According to International Patent Classification (IPC) or to both national classification and IPC

## B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/601, 395/617; 395/200.39, 395/200.47, 395/200.77; 463/40, 463/41, 463/42, 463/43; G06F 17/30

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS  
IEEE Electronic Database

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, P	US 5,561,769 A, (KUMAR et al.), 01 OCTOBER 1996, col. 3, lines 35-46, col. 1, lines 41-52, col. 3, lines 49-57.	1, 3-4, 8-9, 12-14, 25-26, 31, 38-39, 50-52, 56, 59
Y	US 4,570,930 A (MATHESON), 18 FEBRUARY 1986, abstract, background of the invention, col. 3, lines 20-52, col. 4, lines 62-68.	1, 3-4, 8-9, 12-14, 25-26, 31, 38-39, 50-52, 56, 59
A, P	US 5,586,257 A (PERLMAN), 17 DECEMBER 1996, abstract, description of related art, col. 9, lines 21-64	

Further documents are listed in the continuation of Box C.

See patent family annex.

* Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"E" earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O" document referring to an oral disclosure, use, exhibition or other means		
"P" document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search

15 AUGUST 1997

Date of mailing of the international search report

24 NOV 1997

Name and mailing address of the ISA/US  
Commissioner of Patents and Trademarks  
Box PCT  
Washington, D.C. 20231

Facsimile No. (703) 305-3230

Authorized officer

WILLIAM TREAT

Telephone No. (703) 305-9699

This Page Blank (uspto)